



# Algorithme et programmation

# Notion de prédicat

En programmation, de nombreuses opérations sont dépendantes de

*l'évaluation d'une condition booléenne,*

c'est à dire de la valeur d'un *prédicat*.

# Notion de **prédicat**

Pour exprimer un prédicat, une **comparaison** de deux éléments de **même type** est très souvent effectuée.

Les **prédicats** élémentaires utilisent les signes :

$==, <, <=, >, >=, !=$

Exemple:

la variable v1 a pour valeur 4

la variable v2 a pour valeur 20

Prédicat	Valeur
$v1 == v2$	
$v1 > 2$	
$v1 < v2$	
$'a' < 'b'$	
$"Pomme" < "Prune"$	

# Notion de prédicat

Il est possible d'utiliser des **prédicats composés**, lorsqu'une comparaison simple ne suffit pas à définir une situation. Pour cela, on utilise des **connecteurs** tel que:

non

et

ou

Exemple:

(indice1 == 1 ou valeur == 2) et coefficient >3

Les parenthèses sont-elles nécessaires ?

# Notion de prédicat

Attention au langage courant.

Par exemple la proposition suivante:

« Matthieu mesure entre 160 et 180 cm » admet pour réponse "vrai" ou "faux" mais ne peut pas s'écrire ainsi en informatique.

- Donnez le prédicat correct.

- Donner la condition pour fermer la fenêtre et dont la condition d'ouverture est donnée ci-après : il fait trop chaud ET il ne pleut pas

# Schéma conditionnel

## Si ... Alors...FinSi

```
<action a>  
Si <prédicat p>  
Alors  
    <action b>  
FinSi  
<action c>
```

- Indiquez la séquence de code réalisée par le processeur selon la valeur de prédicat p(vrai ou faux).
- Donnez un algorithme indiquant l'état de l'eau selon sa température (solide, liquide ou gazeuse) de la manière la plus efficace possible.

# Schéma conditionnel

## Si ... Alors...Sinon...FinSi

```
<action a>  
Si <prédicat p>  
Alors  
    <action b>  
Sinon  
    <action c>  
FinSi  
<action d>
```

- Indiquez la séquence de code réalisée par le processeur selon la valeur de prédicat p.

# Généralisation du schéma conditionnel

Cas <VariableChoix> Parmi

‘1’ : action 1

‘2’ : action 2

‘3’ : action 3

...

‘n’ : action n

Par défaut : action défaut

FinCas



# Structures répétitives

## *Test en fin de boucle*

Répéter

...

...

Jusqu'à <prédicat p>

Faire

...

...

Tant Que <prédicat q>

***peut-on faire correspondre les prédicats p et q et dans ce cas, quelle est la correspondance entre les deux prédicats ?***

# Structures répétitives

## *Test en début de boucle*

Tant Que <prédicat  $p$ >

...

FinTantQue

# Structures répétitives

## *Boucles définies*

Pour Indice Allant de ValeurInitiale  
                                  à ValeurFinale pas ValeurDuPas

...

...

FinPour

# Structures répétitives

## *Boucles définies*

- *Donnez un algorithme permettant d'afficher la table de multiplication de 7 (de 1x7 à 10x7).*
- *Donnez un algorithme permettant d'afficher la table de multiplication d'une valeur saisie par un utilisateur.*
- *Idem mais l'utilisateur donne également les bornes de début et de fin.*

# Structures répétitives

Un utilisateur doit acquérir une série de notes positive.  
La saisie est terminée par l'entrée d'un nombre négatif.

- Indiquez la boucle choisie et justifiez votre choix.